We Claim:

1. A method for designing an application, comprising:

(a) receiving metadata and a policy;

(b) dynamically constructing a user-interface in accordance with the policy; and

(c) creating the application through the user-interface.

2. The method of claim 1, wherein the user interface supports a design surface with a toolbox and wherein the toolbox has a plurality of available components.

3. The method of claim 2, wherein (c) comprises:

(i) creating a representation of the application, the representation having a stage, the stage having at least one component selected from the plurality of available components of the toolbox.

4. The method of claim 2, wherein (c) comprises:

(i) creating a representation of the application, the representation having a stage.

5. The method of claim 3, wherein the representation is displayed in a graphical format.

6. The method of claim 1, wherein (c) comprises:

(i) creating a representation of the application, the representation having a stage, the stage having at least one component.

7. The method of claim 6, wherein (b) comprises:

(i) categorizing each component to one of a plurality of stages.

8. The method of claim 6, wherein the stage includes a first component and a second component, and wherein (b) comprises:

(i) determining an ordering of the first component and the second component.

9. The method of claim 6, wherein (b) comprises:

(i) determining a cardinality of the stage.

10. The method of claim 6, wherein one of the at least one component is associated with a plurality of properties.

11. The method of claim 10, wherein (c) further comprises:

(ii) selecting one of the plurality of properties.

12. The method of claim 6, wherein (b) comprises:

(i) discovering the at least one component that resides on a computer, the computer supporting the user-interface.

13. The method of claim 6, wherein (c) further comprises:

(ii) compiling the representation of the application in concert with the policy.

14. The method of claim 13, wherein the representation of the application is expressed as an extensible markup language (XML) file.

15. The method of claim 13, wherein (c) further comprises:

(iii) in response to (ii), executing a plurality of computer-executable instructions.

16. The method of claim 13, wherein (c) further comprises:

(iii) determining whether an error exists in the representation.

17. The method of claim 16, wherein (c) further comprises:

(iv) in response to (iii), indicating a determined component and a determined stage corresponding to the error.

18. The method of claim 6, wherein the stage is associated with a plurality of components, and wherein (c) further comprises:

(ii) selecting a matched component from the plurality components, the matched component first matching a document being processed.

19. The method of claim 6, wherein the stage is associated with a plurality of components, and wherein (c) further comprises:

(ii) determining whether the plurality of components shall be sequentially ordered.

20. The method of claim 1, wherein (c) comprises:

(i) receiving a command from the user:

(ii) in response to (i), indicating whether the command corresponds to a permitted operation for manipulating a representation of the application.

21. The method of claim 1, wherein (a) comprises:

(i) selecting the policy from a plurality of policies.

22. A computer-readable medium having computer-executable instructions for performing the method recited in claim 1.

23. A computer-readable medium having computer-executable instructions for performing the method recited in claim 3.

24. A computer-readable medium having computer-executable instructions for performing the method recited in claim 12.

25. A computer-readable medium having computer-executable instructions for performing the method recited in claim 18.

26. A computer-readable medium having computer-executable instructions for performing the method recited in claim 19.

27. A system for designing an application, comprising:

a policy module that stores metadata, the metadata representing a set of rules that is associated with the application;

a user-interface module that generates a design surface;

a composition logic module that receives the metadata from the policy module and that restrains the design surface to be consistent with the metadata when displaying a representation of the application through the user-interface module; and

an input module that receives a command from a user to manipulate the design surface and that updates the design surface, through the composition logic module, in accordance with the command.

28. The system of claim 27, wherein the user-interface module comprises a display interface to a video display device, the video display device showing the design surface to the user.

29. The system of claim 27, further comprising:

a complier module that is coupled to the policy module and that transforms the representation into a set of computer-executable instructions, the set of computer-executable instructions being consistent with the metadata contained in the policy module.

30. The system of claim 29, further comprising:

an execution engine that executes the set of computer-executable instructions.

31. The system of claim 27, further comprising:

a memory that stores software, the software supporting a component, wherein the composition logic module discovers the component and provides a display indicator that is associated with the component.

32. The system of claim 27, wherein the policy module is co-located with the user-interface module.

33. The system of claim 27, wherein the policy module is remotely located from the user-interface module.

34. A computer-readable medium having stored thereon a data structure, comprising:

(a) a first data field that contains a first identifier for a first component, the first component being applicable for an application;

(b) a second data field that contains a second identifier for a stage that is associated with the first component; and

(c) a third data field that represents at least one property that is associated with the first component.

35. The computer-readable medium of claim 34, further comprising:

(d) a fourth data field that contains another identifier for another component that is capable of being coupled to the first component.

36. A computer-readable medium having stored thereon a data structure, comprising:

(a) a first data field that contains a first identifier of a first stage for a user-interface;

(b) a second data field that contains a first indicator that indicates a first position of the first stage within a design surface;

(c) a third data field that contains another identifier of another stage for the user-interface; and

(d) a fourth data field that contains another indicator that indicates a second position of the other stage within the design surface.

37. The computer-readable radium of claim 36, further comprising:

(e) a fifth data field that contains an processing indicator that indicates an ordering of a plurality of components that are associated with the first stage.

38. A method for designing an application, comprising:

(a) receiving metadata that is contained in a policy;

(b) dynamically constructing a user-interface in accordance with the policy, the user-interface supporting a design surface and a toolbox with a plurality of available components;

(c) creating a representation of the application, the representation having at least one stage, each stage having at least one component selected from the plurality of available components by a user;

(d) compiling the representation of the application in concert with the policy; and

(e) in response to (d), executing a set of computer-executable instructions.